

REMARKS

Claims 1-27 are currently pending in the subject application and are presently under consideration. Claims 1, 11, 14, 17, 20 and 23 have been amended as shown at pages 2-6 of the Reply. Claims 7-10, 15-16, 19, 21 and 24 have been canceled.

Applicants' representative thanks the Examiner for the courtesies extended during the teleconference of July 22, 2008.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claims 17 and 22 Under 35 U.S.C. §102(a)

Claims 17 and 22 stand rejected under 35 U.S.C. §102(a) as being unpatentable over Murakami *et al.* (US 2003/0167423). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Murakami *et al.* does not teach each and every element of the subject invention as recited in the subject claims.

The subject claims relate to maintaining fine grained information regarding test versions and their relationship to versions of source code in such a way as to allow for robust management reporting. In particular, independent claim 17 recites a test management method comprising: *retrieving metadata regarding test version information in relation to software code version under test; persisting the metadata to a markup language file versioned with test assets and source code; continuously modifying test information such that new features are added and/or removed to test version changes to the software code under test, wherein the file is an XML file; generating test results that are tagged with test version information in relation to software code version under test, the test results and test version information are all version tagged data and dependent on the versions of the software code under test; and transforming the XML file utilizing XSLT to enable management operations to be performed on the data including at least one of selection, query, reporting, suit composition, and scheduling.*

Murakami *et al.* does not teach or suggest the aforementioned novel features as recited in the subject claims. The cited reference discloses a consistency test program. The program supports a consistency test of a machine code file with respect to the original version and current version of its source file. The program product causes the computer system to perform the following: producing a source file attribute record when a source code in a source file is compiled into a machine code

file, and adding the source file attribute record to the machine code file. (See pg. 1, paragraph [0010]).

In contrast, applicants' claimed subject matter discloses an application test management system and a build drop component. The build drop component comprises the executable version of the software under test. During the development process code is continuously modified such that new features are added and/or removed. Thus, test case components need to change to test the changes in the source under test (SUT). Test case component generates test results that correspond to the results of the test as executed on the current version of SUT. Version component monitors and records changes to the SUT component. Accordingly, the test results and the version component changes are all version tagged data, meaning that they are all dependent on the version of the software under test.

Further, test case file component receives or retrieves version data from version component regarding particular source code and tests, and stores them to a file such as an XML (eXtensible Markup Language) file. The XML file can store metadata associated with tests and source code, and can contain all the attributes necessary for query and management including but not limited to pointers to the source under test, requirements under test, configuration under test, and other aspects necessary for filtering. The XML file data can then be loaded into memory or treated like a database utilizing XSLT transformation, for instance, in order to provide management operations including but not limited to selection, query, reporting, suite composition and scheduling. (See pg. 6, lines 11-25 and pg. 8, line 22-pg. 9, line 2).

Murakami *et al.* merely discloses producing a source file attribute record when a source code in a source file is compiled into a machine code file. The source file attribute record includes file location information and information that is to be updated each time the source file is modified. Murakami *et al.* does not disclose a test case file component that receives or retrieves version data from a version component regarding particular source code and tests, and stores them to an XML file. The XML file data can then be loaded into memory or treated like a database utilizing XSLT transformation, in order to provide management operations including but not limited to selection, query, reporting, suite composition and scheduling.

Accordingly, Murakami *et al.* is silent with regard to a test management method, comprising: *..., the test results and test version information are all version tagged data and dependent on the versions of the software code under test; and transforming the XML file*

utilizing XSLT to enable management operations to be performed on the data including at least one of selection, query, reporting, suit composition, and scheduling.

In view of the foregoing, applicants' representative respectfully submits that the cited references fail to teach or suggest all limitations of the claims 17 and 22. Accordingly, withdrawal of this rejection is respectfully requested.

II. Rejection of Claims 1-5, 9, 14 and 18 Under 35 U.S.C. §103(a)

Claims 1-5, 9, 14 and 18 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Murakami *et al.* in view of Cowan (US 7,243,337). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Murakami *et al.* and Cowan, individually or in combination, do not teach each and every element of the subject invention as recited in the subject claims.

The subject claims relate to maintaining fine grained information regarding test versions and their relationship to versions of source code in such a way as to allow for robust management reporting. In particular, independent claim 1 recites an application test management system comprising: *a version component that monitors source under test components and test components for changes; and a test case file component that includes metadata associated with test components and source under test components received from the version component that indicates relationships between versions of source under test components and versions of test cases, the test case file component includes attributes necessary for query and test management and is continuously modified such that new features are added and/or removed to test changes in the source under test components, source under test components represent specific versions of source code; a build drop component that comprises an executable version of the software under test and includes changed data from the version component; and a test catalog that provides a repository for a collection of test case files, test cases, test variations, and namespace metadata, the test catalog is constructed from aggregation of individual test case files which relate to each other in a hierarchical fashion; wherein the test case file component generates test results that are tagged with the versions of the source under test components and saved to a data store for historical analysis, the test results and version component are all version tagged data and dependent on the versions of the software under test; wherein the test case file component is an XML document and is loaded into memory or treated as a database and wherein XSLT is employed to facilitate*

management operations including at least one of query, reporting, suite composition and scheduling; and wherein the test case file component is located in the source file under test and is loaded into the test catalog.

As stated *supra*, Murakami *et al.* does not teach or suggest the aforementioned novel features as recited in the subject claims. The cited reference discloses a consistency test program. The program supports a consistency test of a machine code file with respect to the original version and current version of its source file. The program product causes the computer system to perform the following: producing a source file attribute record when a source code in a source file is compiled into a machine code file, and adding the source file attribute record to the machine code file. (See pg. 1, paragraph [0010]).

In contrast, applicants' claimed subject matter discloses an application test management system and a build drop component. The build drop component comprises the executable version of the software under test. During the development process code is continuously modified such that new features are added and/or removed. Thus, test case components need to change to test the changes in the source under test (SUT). Test case component generates test results that correspond to the results of the test as executed on the current version of SUT. Version component monitors and records changes to the SUT component. Accordingly, the test results and the version component changes are all version tagged data, meaning that they are all dependent on the version of the software under test.

Further, test case file component receives or retrieves version data from version component regarding particular source code and tests, and stores them to an XML file. The XML file can store metadata associated with tests and source code, and can contain all the attributes necessary for query and management including but not limited to pointers to the source under test, requirements under test, configuration under test, and other aspects necessary for filtering. The XML file data can then be loaded into memory or treated like a database utilizing XSLT transformation, for instance, in order to provide management operations including but not limited to selection, query, reporting, suite composition and scheduling. (See pg. 6, lines 11-25 and pg. 8, line 22-pg. 9, line 2).

Murakami *et al.* merely discloses producing a source file attribute record when a source code in a source file is compiled into a machine code file. The source file attribute record includes file location information and information that is to be updated each time the source file is modified. Murakami *et al.* does not disclose a test case file component that receives or retrieves version data

from a version component regarding particular source code and tests, and stores them to an XML file. The XML file data can then be loaded into memory or treated like a database utilizing XSLT transformation, in order to provide management operations including but not limited to selection, query, reporting, suite composition and scheduling.

Cowan does not cure the deficiencies of Murakami *et al.* Cowan discloses a method executed in a computer system for automatically tracking platform usage. One or more records of platform information are recorded for a plurality of program executions. Each of the one or more records of platform information corresponds to one of the plurality of program executions. Data relating to the one or more records of platform information are summarized to assess the effectiveness of platform usage. (*See* col. 2, lines 33-41).

As stated *supra*, applicants' claimed subject matter disclose a test case file component that receives or retrieves version data from a version component regarding particular source code and tests, and stores them to an XML file. The XML file data can then be loaded into memory or treated like a database utilizing XSLT transformation, in order to provide management operations including but not limited to selection, query, reporting, suite composition and scheduling. Cowan merely discloses storing distinguishing information enabling one to identify a particular version of the software component in a library. Accordingly, Cowan is silent with regard to an application test management system, ... ***wherein the test case file component is an XML document and is loaded into memory or treated as a database and wherein XSLT is employed to facilitate management operations including at least one of query, reporting, suite composition and scheduling; and wherein the test case file component is located in the source file under test and is loaded into the test catalog.***

Further, independent claim 14 recites a test management system comprising: ***a means for maintaining fine-grained track of a test's relation to a version of software under test; a means for querying test data to facilitate generation of test management reports; a means for continuously modifying test data such that new features are added and/or removed to test version changes to the software under test; and a means for generating test results that are tagged with test version data in relation to the version of software under test, the test results and test version data are all version tagged data and dependent on the versions of the software under test; wherein the means for maintaining fine-grained track of a test's relation to a version of software under test includes***

persisting software version information and related test information to an XML file, and wherein the XML file is transformed utilizing XSLT to enable test data to be queried.

As stated *supra*, Murakami *et al.* merely discloses producing a source file attribute record when a source code in a source file is compiled into a machine code file. The source file attribute record includes file location information and information that is to be updated each time the source file is modified. And, Cowan merely discloses storing distinguishing information enabling one to identify a particular version of the software component in a library. Applicants' claimed subject matter discloses a test case file component that receives or retrieves version data from a version component regarding particular source code and tests, and stores them to an XML file. The XML file data can then be loaded into memory or treated like a database utilizing XSLT transformation, in order to provide management operations including but not limited to selection, query, reporting, suite composition and scheduling.

In view of the foregoing, applicants' representative respectfully submits that the cited references fail to teach or suggest all limitations of independent claims 1 and 14 (and claims 2-5, 9 and 18 that depend there from). Accordingly, withdrawal of this rejection is respectfully requested.

III. Rejection of Claims 19-21 Under 35 U.S.C. §103(a)

Claims 19-21 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Murakami *et al.* in view of Mandava *et al.* (US 7,203,928) hereinafter "Mandava-2". It is respectfully requested that this rejection should be withdrawn for at least the following reasons. Murakami *et al.* and Mandava-2, individually or in combination, do not teach or suggest each and every element as set forth in the subject claims. In particular, Mandava-2 does not make up for the aforementioned deficiencies of Murakami *et al.* with respect to independent claim 17 (which claims 19-21 depend from). Thus, the claimed subject matter as recited in claims 19-21 is not obvious over the combination of Murakami *et al.* and Mandava-2.

IV. Rejection of Claims 23-27 Under 35 U.S.C. §103(a)

Claims 23-27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Mandava-2 in view of Murakami *et al.* It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Mandava-2 and Murakami *et al.*, individually or in combination, do not teach each and every element of the subject invention as recited in the subject claims.

Independent claim 23 recites *loading a test case in accordance with a test case file stored in a source file; executing the test case on a source code under test; generating test results, wherein the test results are version tagged to indicate the relationship between test results, version of test case, and version of source code under test; and continuously modifying test information such that new features are added and/or removed to test version changes to the source code under test; continuously modifying test information such that new features are added and/or removed to test version changes to the source code under test; saving test results to an XML file; providing a repository for a collection of test case files, test cases, test variations, and namespace metadata, wherein the repository is constructed from aggregation of individual test case files which relate to each other in a hierarchical fashion; and employing XSLT to facilitate management operations including at least one of query, reporting, suite composition and scheduling.* The subject claim discloses providing information that indicates the relationship between a test result and the version of a test case and version of source under test. Mandava-2 does not teach or suggest the aforementioned novel features as recited in the subject claims. The cited reference discloses a method for providing standardized reporting of test results. The system employs codes that testers can associate expected results and rationale for the results. This allows different testers to understand the test that was created and what is expected when the test is executed. However, Mandava-2 is silent regarding versions of a test case and version of source code being tested. Therefore, Mandava-2 fails to teach or suggest generating test results, wherein the test results are version tagged to indicate the relationship between test results, test case version, and source under test version.

Murakami *et al.* does not cure the deficiencies of Mandava-2. As stated *supra*, Murakami *et al.* discloses a consistency test program. The program supports a consistency test of a machine code file with respect to the original version and current version of its source file. The program product causes the computer system to perform the following: producing a source file attribute record when a source code in a source file is compiled into a machine code file, and adding the source file attribute record to the machine code file. (*See* pg. 1, paragraph [0010]).

In contrast, applicants' claimed subject matter discloses an application test management system and a build drop component. The build drop component comprises the executable version of the software under test. During the development process code is continuously modified such that new features are added and/or removed. Thus, test case components need to change to test the

changes in the source under test (SUT). Version component monitors and records changes to the SUT component. Accordingly, the test results and the version component changes are all version tagged data, meaning that they are all dependent on the version of the software under test.

Further, test case file component receives or retrieves version data from version component regarding particular source code and tests, and stores them to an XML file. The XML file can store metadata associated with tests and source code, and can contain all the attributes necessary for query and management including but not limited to pointers to the source under test, requirements under test, configuration under test, and other aspects necessary for filtering. The XML file data can then be loaded into memory or treated like a database utilizing XSLT transformation, for instance, in order to provide management operations including but not limited to selection, query, reporting, suite composition and scheduling. (See pg. 6, lines 11-25 and pg. 8, line 22-pg. 9, line 2).

Murakami *et al.* merely discloses producing a source file attribute record when a source code in a source file is compiled into a machine code file. The source file attribute record includes file location information and information that is to be updated each time the source file is modified. Murakami *et al.* does not disclose a test case file component that receives or retrieves version data from a version component regarding particular source code and tests, and stores them to an XML file. The XML file data can then be loaded into memory or treated like a database utilizing XSLT transformation, in order to provide management operations including but not limited to selection, query, reporting, suite composition and scheduling.

Accordingly, Murakami *et al.* is silent with regard to a testing methodology, comprising: *...saving test results to an XML file; providing a repository for a collection of test case files, test cases, test variations, and namespace metadata, wherein the repository is constructed from aggregation of individual test case files which relate to each other in a hierarchical fashion; and employing XSLT to facilitate management operations including at least one of query, reporting, suite composition and scheduling.*

In view of the foregoing, applicants' representative respectfully submits that the cited references fail to teach or suggest all limitations of independent claim 23 (and claims 24-27 that depend there from). Accordingly, withdrawal of this rejection is respectfully requested.

V. Rejection of Claims 7-8, 10-12 and 15-16 Under 35 U.S.C. §103(a)

Claims 7-8, 10-12 and 15-16 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Murakami *et al.* in view of Cowan and further in view of Mandava-2. It is respectfully requested that this rejection should be withdrawn for at least the following reasons. Murakami *et al.*, Cowan and Mandava-2, individually or in combination, do not teach or suggest each and every element as set forth in the subject claims. In particular, Mandava-2 does not make up for the aforementioned deficiencies of Murakami *et al.* and Cowan with respect to independent claims 1 and 14 (which claims 7-8, 10-12 and 15-16 depend from). Thus, the subject invention as recited in claims 7-8, 10-12 and 15-16 is not obvious over the combination of Murakami *et al.*, Cowan and Mandava-2.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP641US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/Marisa J. Zink/

Marisa J. Zink

Reg. No. 48,064

AMIN, TUROCY & CALVIN, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731